

グループワークによるソフトウェア開発における成果物管理支援

荷田 雄太[†] 栗原 紘樹[†] 松浦 佐江子[†]

芝浦工業大学 システム工学部電子情報システム学科[†]

1. はじめに

本学の電子情報システム学科 3 年の後期にある情報実験 では6~9人のグループに分かれてJavaプログラミングを用いたソフトウェア開発を行っている[1]。本実験では開発工程を要求分析、システム分析、システム設計、実装・テストフェーズに分けており、2003 年度よりグループワーク支援システムを用いてグループ作業を支援している[1]。支援内容は掲示板を用いた連絡・議論の支援、作業状況を把握するための作業報告書の作成、フェーズごとの成果物のアップロードである。実験では、大量のドキュメントファイル、ソースコード、テストプログラムが作成され、現在の支援システムでは成果物のアップロードによりグループでのファイル共有を支援している。しかし、ファイルを整理し、効率よく管理するための支援が行われていないため、ファイル共有の仕組みを有効活用できていない。特に問題が起きているのが実装フェーズであり、テストを満足に済ませることはおろか、自分達のプログラムがどこまで動作するのかを把握しきれない学生がいるというのが現状である。

2. グループワーク支援システム概要

グループワーク支援システムとは、[2][3]で共同開発されたWEBアプリケーションで、機能として作業計画書管理、作業報告書管理、論議内容管理、アドレス帳管理、教員・管理者との連絡・質疑応答管理、仕様書管理、インスペクション支援がある。作業計画書は、作業計画を立ててグループのメンバがその計画に基づき行動することで、メンバの作業状況や自分のすべきことの把握など円滑なグループワークによるソフトウェア開発をサポートする狙いがある。学生は、指針としてあらかじめ与えられている各フェーズで行うべき作業の一覧から作業項目を選択して作業計画を立てる。作業計画を立てると各作業項目に対応した掲示板とアップローダが生成される。学生は作業の中で作成される成果物をアップローダによって共有し、掲示板を使って成果物に関する問題を議論する。支援システムで管理する仕様書は「要求仕様書」「システム仕様書」「システム設計書」「ソースコード」「操作説明書」「テスト仕様書」「テスト結果報告書」である。システム内では仕様書も成果物として扱われる。

3. 支援システムでの成果物管理における問題点

3.1 ファイルアップローダの問題点

学生は作業計画書に従って作業項目に対応する作業を進めていく。作業項目ごとにアップローダが作成されるわけだが、グループワークが進み、作業項目が増えくるとアップローダの数が多くなり、目的のアップローダを探しにくくなるという問題がある。この原因として、アップローダが仕様書単位で、作業項目が作成された順に一覧表示

されることが挙げられる。仕様書を作成するためにはユースケース図、ユースケース記述、ワークフローといった数種類の成果物を作成することになるので、ドキュメントの種類に関わらず一覧表示をするのでは種類を混同して表示してしまい、把握が難しい。成果物を閲覧するためのアップローダが整理されていないことが、グループで成果物を共有する障害となる。

3.2 ファイルの管理方法

支援システム上では、「ユースケース記述の作成」や「シーケンス図の作成」といった作業項目単位でファイルを管理することになる。しかし、成果物の種別に関わらず時系列でバージョン管理するのみであり、例えば「コーディング」と「統合テストの定義」といった関連した作業項目が別になっているために、ソースコードとテストプログラムが関連付けた方式で各ソースコードのバージョンを管理することができなかった。そのため、テストに漏れが生じ、ソフトウェアの品質を落とす原因となっていた。成果物単位でバージョン管理することができればソースコードごとのバージョン管理が可能になる。

4. 新しい成果物管理支援方法の提案

4.1 アップローダのカテゴリ化

今まではアップローダを仕様書単位で一覧表示させていたが、仕様書の中で、さらに成果物の種類をカテゴリとして分けることで表示を分けるようにする。カテゴリ化とはフェーズの作業項目を「方向付け」、「定義」、「検査」、「まとめ」にわけを示す。各仕様書は図1のような手順でカテゴリごとに指定された成果物をまとめることで作成される。閲覧・投稿したい成果物の種類を特定することで、不必要なアップローダを表示させないようにする。また、同じカテゴリに含まれる作業項目の成果物の状況も把握しやすくする。

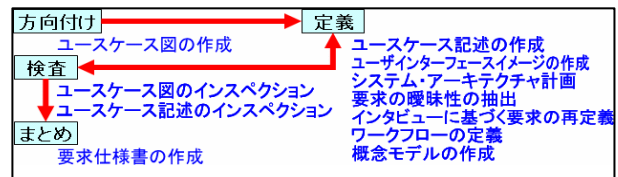


図1：要求仕様書のカテゴリと成果物

4.2 成果物単位でのバージョン管理

現在、成果物のバージョン管理はアップローダ単位に時系列で管理しているだけである。アップローダ単位では担当者ごとの成果物、ファイル形式の違い、代替案などでのバージョン管理ができないため不十分と考える。任意の成果物単位に時系列でのバージョン管理を行えるようにする。この機能によって、作業項目でどのような成果物を作成しているのか、成果物がどのように推移していったのかを把握しやすくする。仕様書を作成するときには各成果物の最新版を参照することになるため、最新版を取得するのが容易にする必要がある。

4.3 パッケージ構成を維持したソースコード管理

3.2 節で述べたように、今までの管理方法ではソースコ

Documentation Management System for Group Work Software Development

[†] Yuta Hasuda, Hiroki Kurihara, Saeko Matuura

[†] Shibaura Institute of Technology Department of Electronic Information Systems

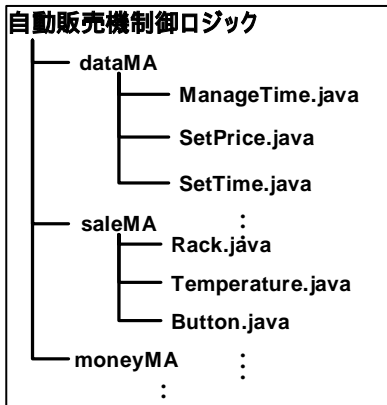


図2：ディレクトリ構造での管理

は複数人が分散して作成するため、パッケージ構成を確認しながら投稿・バージョン管理することで実装時の進捗具合の共有を促す。図2は昨年度の情報実験 II で開発されたシステムの一部を本機能に適用した例である。昨年は「コーディング」という作業項目一つに図2のJavaファイルがクラス、バージョンが管理されずに投稿されていた。

4.4 パッケージ構成の履歴保存

実装作業ではクラスの変更が頻繁に発生すると考えられる。一つのクラスの変更により、エラーが発生するという事態も起こるため、パッケージ構成とそれに含まれるクラスを保存できるようにする。そうすることで、動作が保障されている構造への後戻りを行えるようにする。パッケージ構成の保存についてはSubversion[4]の仕組みを参考に、グループワーク支援システムに適用する。

5. 有効性の評価

情報実験 のシステム設計フェーズ終了時に学生に行ったアンケートの結果を以下に示す。(小数点以下は2桁で四捨五入、履修者は69名だが集計は回答者62名を全体の人数として行う)

質問1：今年度のアップローダは「方向付け」、「定義」、「検査」、「まとめ」というカテゴリによって区別されていました。成果物を投稿しようとしたときに投稿先のアップローダの位置を探すのは難しかったですか？	
難しかった	簡単だった
24(38.70%)	38(61.29%)

質問2：質問1で「難しかった」と答えた人は難しかった理由を教えてください。		
レイアウト	カテゴリで区別されていた	その他
18(75.0%)	6(25.0%)	0(0%)

質問3：質問1で「簡単であった」と答えた人は簡単であった理由を教えてください。		
レイアウト	カテゴリで区別されていた	その他
9(23.68%)	29(76.32%)	0(0%)

質問より、学生の6割以上がアップローダを探しやすかったと思っていることがわかった。4.1節で提案している「アップローダのカテゴリ化」は簡単であった理由の7割強を締めているため有効であったと考えられる。しかし、レイアウトの問題からアップローダを探し辛いと感じている学生が全体の29%にも及んでいる。システムのレイアウトを改善することでさらなる有効な支援が期待できる。

ード同士の関連を理解することはできなかった。情報実験 ではJavaプログラミング言語を用いている。従って、クラスの配置を表すパッケージ構成を維持した状態での成果物管理方法を提案する。テストプログラムも同じように、構造を維持した管理方法にする。ソースコード

質問4：一つのアップローダでは複数の成果物のバージョン管理を行うことができます。アップロードするときに複数のバージョン管理をする機会はありましたか？	
はい	いいえ
29(46.77%)	33(53.23%)

成果物単位のバージョン管理に関しては利用していない学生が半分以上であったことが判明した。次の質問で利用者がどのような場合に使用したのかを検証する。

質問5：質問4で「はい」と答えた人はどのような場合に使ったかを教えてください。		
担当者別	ファイル拡張子別	その他
19(65.52%)	4(13.79%)	6(20.69%)

利用した学生の約8割は4.2節で提案した使用をしている。その他と答えた学生の使用方法は次のようになっている。
 1. クラス図を投稿するときにファイル形式で分けた。
 2. インスペクションをして変更した場合は分けて管理し、細かい修正はバージョンアップをした。
 3. 一つのアップローダで作成すべき成果物が複数あった場合に使用した。システムのログを閲覧したところ、「シーケンス図を作成する」の作業項目では基本フローと例外フローを区別してバージョン管理している班も見られた。また、独自のルールで成果物は担当者別に投稿すると決め、ファイル名に担当者バージョンをつけて管理する班もあった。ファイル名への制約は与えていないのだが、名前を見ただけで担当とバージョンが分かる支援もわかりやすいと感じた。投稿するときにファイル名を迷わないで済み、かつ閲覧時に成果物の情報がファイル名からわかるためである。

質問6：仕様書をまとめるときにはフェーズで作成してきた成果物の最新版を参照することになると思います。作成時に適切な成果物を参照することは簡単でしたか？		
仕様書を作成していない	難しかった	簡単だった
26(41.94%)	8(12.9%)	28(45.16%)

仕様書を作成した学生の77.8%が成果物の最新版を探しやすいと感じていることがわかる。学生の大半が成果物の位置がわかりやすいためと答えており、提案4.2が有効であったと考えられる。

6. まとめ

アンケート結果とシステムのログより、昨年のアップローダに比べ、管理方法に幅が出たことがわかる。4章で提案した機能の内「アップローダのカテゴリ化」「成果物単位のバージョン管理」が学生に対し、有効な支援をしていると考えられる。実験が終了した後、再びアンケートを実施し、実装フェーズ以降の意見を取得・分析する。また、各グループの成果物のログを分析することで、昨年と比較してソフトウェアの品質が向上しているかを検証する。

7. 参考文献

[1] S. Matsuura: Practical Software Engineering Education based on Software Development Group Experiments, E-Learn2005 pp.1701-1708, 2005.
 [2] 青沼、松浦: グループワークによるソフトウェア開発教育のための進捗状況支援システム、情報処理学会第66回全国大会、pp.237-238、2004
 [3] 吉田、松浦: グループプログラミング授業における仕様書作成支援ツールの研究、情報処理学会第66回全国大会、pp.237-238、2004
 [4] Subversionによるバージョン管理
<http://subversion.bluegate.org/doc/book.html>